User AF

PTO/SB/21 (09-04)
Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Total Number of Pages in This Submission

Application Number	09/683,351
Filing Date	February 18, 2001
First Named Inventor	Leber
Art Unit	2183
Examiner Name	Kevin P. Rizzuto
Attorney Docket Number	DE920000097US1 (8728-729)

ENCLOSURES *(Check all that apply)*

- [X] Fee Transmittal Form
  - [ ] Fee Attached
- [ ] Amendment/Reply
  - [ ] After Final
  - [ ] Affidavits/declaration(s)
- [ ] Extension of Time Request
- [ ] Express Abandonment Request
- [ ] Information Disclosure Statement
- [ ] Certified Copy of Priority Document(s)
- [ ] Reply to Missing Parts/ Incomplete Application
  - [ ] Reply to Missing Parts under 37 CFR 1.52 or 1.53
- [ ] Drawing(s)
- [ ] Licensing-related Papers
- [ ] Petition
- [ ] Petition to Convert to a Provisional Application
- [ ] Power of Attorney, Revocation Change of Correspondence Address
- [ ] Terminal Disclaimer
- [ ] Request for Refund
- [ ] CD, Number of CD(s) _____
  - [ ] Landscape Table on CD
- [ ] After Allowance Communication to TC
- [ ] Appeal Communication to Board of Appeals and Interferences
- [X] Appeal Communication to TC **(Appeal Notice, Brief, Reply Brief)**
- [ ] Proprietary Information
- [ ] Status Letter
- [ ] Other Enclosure(s) (please Identify below):

Remarks

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm Name	F. Chau & Associates, LLC		
Signature			
Printed name	Koon Hon Wong		
Date	November 14, 2005	Reg. No.	48,459

CERTIFICATE OF TRANSMISSION/MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below:

Signature			
Typed or printed name	Koon Hon Wong	Date	November 14, 2005

PTO/SB/17 (12-04)
Approved for use through 07/31/2006. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Effective on 12/08/2004.
Fees pursuant to the Consolidated Appropriations Act, 2005 (H.R. 4818).

FEE TRANSMITTAL For FY 2005

☐ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT ($) 500.00

Complete if Known	
Application Number	09/683,351
Filing Date	February 18, 2001
First Named Inventor	Leber
Examiner Name	Kevin P. Rizzuto
Art Unit	2183
Attorney Docket No.	DE920000097US1 (8728-729)

METHOD OF PAYMENT (check all that apply)

☐ Check ☐ Credit Card ☐ Money Order ☐ None ☐ Other (please identify): _____

☒ Deposit Account Deposit Account Number: 50-0150 Deposit Account Name: IBM/YORKTOWN HEIGHTS

For the above-identified deposit account, the Director is hereby authorized to: (check all that apply)

☒ Charge fee(s) indicated below 50-0510 ☐ Charge fee(s) indicated below, **except for the filing fee**

☒ Charge any additional fee(s) or underpayments of fee(s) under 37 CFR 1.16 and 1.17 ☐ Credit any overpayments

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

FEE CALCULATION

1. BASIC FILING, SEARCH, AND EXAMINATION FEES

Application Type	FILING FEES Fee ($)	FILING FEES Small Entity Fee ($)	SEARCH FEES Fee ($)	SEARCH FEES Small Entity Fee ($)	EXAMINATION FEES Fee ($)	EXAMINATION FEES Small Entity Fee ($)	Fees Paid ($)
Utility	300	150	500	250	200	100	
Design	200	100	100	50	130	65	
Plant	200	100	300	150	160	80	
Reissue	300	150	500	250	600	300	
Provisional	200	100	0	0	0	0	

2. EXCESS CLAIM FEES

Fee Description	Fee ($)	Small Entity Fee ($)
Each claim over 20 or, for Reissues, each claim over 20 and more than in the original patent	50	25
Each independent claim over 3 or, for Reissues, each independent claim more than in the original patent	200	100
Multiple dependent claims	360	180

Total Claims	Extra Claims	Fee ($)	Fee Paid ($)
_____ - 20 or HP =	_____ x	_____ =	_____

HP = highest number of total claims paid for, if greater than 20

Indep. Claims	Extra Claims	Fee ($)	Fee Paid ($)
_____ - 3 or HP =	_____ x	_____ =	_____

HP = highest number of independent claims paid for, if greater than 3

Multiple Dependent Claims Fee ($)	Fee Paid ($)
_____	_____

3. APPLICATION SIZE FEE

If the specification and drawings exceed 100 sheets of paper, the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).

Total Sheets	Extra Sheets	Number of each additional 50 or fraction thereof	Fee ($)	Fee Paid ($)
_____ - 100 =	_____ / 50 =	_____ (round up to a whole number) x	_____ =	_____

4. OTHER FEE(S)

	Fees Paid ($)
Non-English Specification, $130 fee (no small entity discount)	
Other: Appeal Brief	500.00

SUBMITTED BY

Signature		Registration No. (Attorney/Agent) 48,459	Telephone 516-692-8888
Name (Print/Type)	Koon Hon Wong		Date November 14, 2005

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Applicants:	Leber et al.	Examiner: Kevin P. Rizzuto
Serial No:	09/683,351	Group Art Unit: 2183
Filed:	December 18, 2001	Docket: DE920000097US1 (8728-729)
For:	METHOD FOR HANDLING 32 BIT RESULTS FOR AN OUT-OF-ORDER PROCESSOR WITH 64-BIT ARCHITECTURE	

APPEAL BRIEF

This is an Appeal from the Final Office Action mailed June 8, 2005 (Paper No. 20040907), finally rejecting claims 1-10. Applicants appeal pursuant to the Notice of Appeal filed on September 12, 2005 and submit this appeal brief.

Appeal from Group 2183

F. Chau & Associates, LLC
130 Woodbury Road
Woodbury, NY 11797
TEL: (516) 692-8888
FAX: (516) 692-8889
Attorneys for Appellants

TABLE OF CONTENTS

1. Real Party in Interest

The real party in interest is INTERNATIONAL BUSINESS MACHINES CORPORATION, the assignee of the entire right, title and interest in and to the subject application by virtue of an assignment of record.

2. Related Appeals and Interferences

None.

3. Status of Claims

Claims 1-10 are pending, stand rejected and are under appeal.

A copy of the claims 1-10 as pending is presented in the Claims Appendix.

4. Status of Amendments

Claims 1, 8, 9 and 10 were amended in Applicants' Amendment Under 37 C.F.R. § 1.116 mailed on August 8, 2005. The Examiner entered Applicants' Amendment according to Examiner's Advisory Action mailed on August 30, 2005.

5. Summary of Claimed Subject Matter

In independent claim 1, a method for operating a processor having an architecture of a larger bit-length with a program comprising instructions compiled to produce instruction results of at least one smaller bit-length is provided.

It is detected when in program order a first instruction is to be dispatched which does not have a target register address as one of its sources (Fig. 11, #1110; page 16,

paragraph 104). The first instruction is one of the instructions compiled to produce instruction results of at least one smaller bit-length. An extract instruction (Fig. 8; Fig. 9) is added into an instruction stream before the first instruction (Fig. 11, #1130; page 15, paragraph 94). The extract instruction includes the following steps:

(a) the extract instruction is dispatched together with the following first instruction from an instruction queue into a Reservation Station (Fig. 11, #1140; page 16, paragraph 95);

(b) the extract instruction is issued to an Instructional Execution Unit (IEU) (Page 7, paragraph 42) as soon as all source operand data is available and an IEU is available according to respective issue scheme (Fig. 11, #1150, #1160, #1170; page 16, paragraphs 96, 97, 98, 99, 100);

(c) the extract instruction is executed by an available IEU (Fig. 11, #1180; page 16, paragraph 100);

(d) an indication is set that the result of said extract instruction needs to be written into a result field of the first instruction following the extract instruction (Fig. 11, #1190, paragraph 101); and

(e) the extract instruction result is written into the result field of said first instruction, and into all fields of operands being dependent of said first instruction (Fig. 11, #1195; paragraph 102).

In independent claim 8, a computer system having an out-of-order processing system for operating a processor having an architecture of a larger bit-length with a program comprising instructions compiled to produce instruction results of at least one

smaller bit-length is provided. The computer system uses a computer readable machine language. The computer readable machine language includes the following:

(a) a first computer readable code for detecting when in program order a first instruction is to be dispatched which does not have a target register address as one of its sources (Fig. 11, #1110; page 16, paragraph 104). The first instruction is one of the instructions compiled to produce instruction results of at least one smaller bit-length; and

(b) a second computer readable code for adding an extract instruction (Fig. 8; Fig. 9) into an instruction stream before the first instruction (Fig. 11, #1130; page 15, paragraph 94). The extract instruction includes the following additional computer readable code:

(i) a third computer readable code for dispatching the extract instruction together with the following first instruction from the instruction queue into a Reservation Station (Fig. 11, #1140; page 16, paragraph 95),

(ii) a fourth computer readable code for issuing the extract instruction to an Instruction Execution Unit (IEU) as soon as all source operand data is available and an IEU is available according to respective issue scheme (Fig. 11, #1150, #1160, #1170; page 16, paragraphs 96, 97, 98, 99, 100),

(iii) a fifth computer readable code for executing the extract instruction by an available IEU (Fig. 11, #1180; page 16, paragraph 100),

(iv) a sixth computer readable code for setting an indication that the result of said extract instruction needs to be written into a result field of the first instruction following the extract instruction (Fig. 11, #1190, paragraph 101), and

(v) a seventh computer readable code for writing the extract instruction result into the result field of said first instruction, and into all fields of operands being dependent of said first instruction (Fig. 11, #1195; paragraph 102).

In independent claim 9, a computer system having an out-of-order processing system for operating a processor having an architecture of a larger bit-length with a program comprising instructions compiled to produce instruction results of at least one smaller bit-length is provided. The computer system executes a readable machine language. The readable machine language includes the following:

(a) a first computer readable code for detecting when in program order a first instruction is to be dispatched which does not have a target register address as one of its sources (Fig. 11, #1110; page 16, paragraph 104). The first instruction is one of the instructions compiled to produce instruction results of at least one smaller bit-length; and

(b) a second computer readable code for adding an extract instruction (Fig. 8; Fig. 9) into an instruction stream before the first instruction (Fig. 11, #1130; page 15, paragraph 94). The extract instruction includes the following additional computer readable code:

(i) a third computer readable code for dispatching the extract instruction together with the following first instruction from the instruction queue into a Reservation Station (Fig. 11, #1140; page 16, paragraph 95),

(ii) a fourth computer readable code for issuing the extract instruction to an Instruction Execution Unit (IEU) as soon as all source operand data is available and an IEU is available according to respective issue scheme (Fig. 11, #1150, #1160, #1170; page 16, paragraphs 96, 97, 98, 99, 100),

(iii) a fifth computer readable code for executing the extract instruction by an available IEU (Fig. 11, #1180; page 16, paragraph 100),

(iv) a sixth computer readable code for setting an indication that the result of said extract instruction needs to be written into a result field of the first instruction following the extract instruction (Fig. 11, #1190, paragraph 101), and

(v) a seventh computer readable code for writing the extract instruction result into the result field of said first instruction, and into all fields of operands being dependent of said first instruction (Fig. 11, #1195; paragraph 102).

In independent claim 10, a processing system having an out-of-order processing system for operating a processor having an architecture of a larger bit-length with a program comprising instructions compiled to produce instruction results of at least one smaller bit-length is provided. The processing system has means for executing a readable machine language. The readable machine language includes the following:
(a) a first computer readable code for detecting when in program order a first instruction is to be dispatched which does not have a target register address as one of its sources (Fig. 11, #1110; page 16, paragraph 104). The first instruction is one of the instructions compiled to produce instruction results of at least one smaller bit-length; and

(b) a second computer readable code for adding an extract instruction (Fig. 8; Fig. 9) into an instruction stream before the first instruction (Fig. 11, #1130; page 15, paragraph 94). The extract instruction includes the following additional computer readable code:

(i) a third computer readable code for dispatching the extract instruction together with the following first instruction from the instruction queue into a Reservation Station (Fig. 11, #1140; page 16, paragraph 95),

(ii) a fourth computer readable code for issuing the extract instruction to an Instruction Execution Unit (IEU) as soon as all source operand data is available and an IEU is available according to respective issue scheme (Fig. 11, #1150, #1160, #1170; page 16, paragraphs 96, 97, 98, 99, 100),

(iii) a fifth computer readable code for executing the extract instruction by an available IEU (Fig. 11, #1180; page 16, paragraph 100),

(iv) a sixth computer readable code for setting an indication that the result of said extract instruction needs to be written into a result field of the first instruction following the extract instruction (Fig. 11, #1190, paragraph 101), and

(v) a seventh computer readable code for writing the extract instruction result into the result field of said first instruction, and into all fields of operands being dependent of said first instruction (Fig. 11, #1195; paragraph 102).

6. Grounds of Rejection to be Reviewed on Appeal

A. Claims 1-10 stand rejected under 35 U.S.C. § 112, second paragraph as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

B. Claims 1-5 and 7-10 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Mahurin et al. (U.S. Patent No. 6,493,819) (hereinafter "Mahurin") in view of Isaman (U.S. Patent No. 6,449,710) (hereinafter "Isaman"), and Superscalar Microprocessor Design by Johnson (hereinafter "Johnson") and Computer Organization and Design by Hennessy and Patterson (hereinafter "Hennessy").

7. Argument

A. Introduction

The law of indefiniteness was recently revisited on August 5, 2005 by the Court of Appeals of the Federal Circuit (CAFC) in *Datamize, LLC v. Plumtree Software, Inc.*, 417 F.3d 1342 (Fed. Cir. 2005):

> According to the Supreme Court, "[t]he statutory requirement of particularity and distinctness in claims is met only when [the claims] clearly distinguish what is claimed from what went before in the art and clearly circumscribe what is foreclosed from future enterprise." The definiteness requirement, however, does *not* compel absolute clarity. Only claims "not amenable to construction" or "insolubly ambiguous" are indefinite. Thus, the definiteness of claim terms depends on whether those terms can be given *any* reasonable meaning.

Datamize,LLC, 417 F.3d at 11-12 (citations omitted) (emphasis added). It is respectfully submitted that the instant claims meet the statutory requirement of particularity and distinctness. The instant claims are amenable to construction and are not insolubly ambiguous. For the reasons set forth below, Appellants respectfully request that the claim rejections under 35 U.S.C. § 112, second paragraph, be reversed.

In rejecting claims under 35 U.S.C. § 103, the Examiner bears the burden of presenting a *prima facie* case of obviousness. *In re Rijckaert*, 9 F.3d 1531, 1532 (Fed. Cir. 1993). The burden of presenting a *prima facie* case of obviousness is only satisfied by showing some objective teaching in the prior art or that knowledge generally available to one of ordinary skill in the art would lead that individual to combine the relevant teachings of the references. *In re Fine*, 837 F.2d 1071, 1074 (Fed. Cir. 1988). A *prima facie* case of obviousness is established when the teachings of the prior art itself would appear to have suggested the claimed subject matter to one of ordinary skill in the art. *In re Bell*, 991 F.2d

781, 782 (Fed. Cir. 1993). The suggestion to combine the references should come from the prior art, and the Examiner cannot use hindsight gleaned from the invention itself to pick and choose among related disclosures in the prior art to arrive at the claimed invention. *In re Fine*, 837 F.2d at 1075. If the Examiner fails to establish a *prima facie* case, the rejection is improper and must be overturned. *In re Rijckaert*, 9 F.3d at 1532 (citing In re Fine, 837 F.2d at 1074).

It is respectfully submitted that at the very least, the references of Mahurin, Isaman, Johnson and Hennessy, taken individually or in any combination, are legally insufficient to sustain a *prima facie* case of obviousness against independent claims 1, 8, 9 and 10.

For the reasons set forth below, Appellants respectfully request that the claim rejections under 35 U.S.C. § 103(a) be reversed.

B. Claims 1-10 stand rejected under 35 U.S.C. § 112, second paragraph as being as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

(i). Claims 1-10 meet the statutory requirements of 35 U.S.C. § 112, second paragraph.

The Examiner argues that the claimed "result field of an instruction" has a common meaning that there is a field, i.e., a number of bits, within an instruction stored in the instruction memory. The Examiner further argues that the "result field" points to a location in memory or a registry, and does not actually get a result. In support, the Examiner cites extrinsic evidence – Hennessy.

Whether or not the Examiner is correct as to the "common meaning" of a term is entirely *irrelevent* to the analysis of 35 U.S.C. § 112, second paragraph. It should be noted

that, when the Examiner argues that the "result field" does not get a result, the Examiner is implictly admitting that Appellants provide a meaning for "result field." That is, the Examiner does not dispute that the instant application provides an unambiguous meaning for "result field," which satisfies § 112, second paragraph, *per se*. The Examiner only disagrees with such a meaning in the claims, such as "writing the extract instruction result *into the result field* of the first instruction." This is an improper rejection under 35 U.S.C. § 112, second paragraph.

Irrespective of the above, the instant application does not conflict with Hennessy. Hennessy refers to to a field in an external instruction, whereas the instant application refers to a field in *an internal representation of an instruction in a processor*. This is clear when reading the application, and considering the use of the term "instruction" throughout the application. FIG. 8 provides an exemplary instruction with a number of fields, including "Opcode," "src1," "src2," and "result." Writing into the result field in FIG. 8 is indicated by an arrow pointing into the result field. Writing into the result field is described in detail on p. 11, para. [0062].

Because the claims fully meet the statutory requirement of particularity and distinctness as required by 35 U.S.C. § 112, second paragraph, the rejection of claims 1-10 should be reversed.

C. **Claims 1-5 and 7-10 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Mahurin in view of Isaman, Johnson and Hennessy.**

(i). **The combination of Mahurin, Isaman, Johnson and Hennessy fails to teach or suggest "adding an extract instruction into an instruction stream before the first instruction," as claimed in claims 1, 8, 9 and 10.**

The Examiner contends that col. 5, lines 35-54 of Mahurin teaches "adding an extract instruction into an instruction stream before the first instruction." (Paper no. 20040907, p. 5). Col. 5, lines 35-42 of Mahurin teaches the following:

> MROM instructions are instructions which are determined to be too complex for decode by decode units 20. MROM instructions are executed by invoking MROM unit 34. More specifically, when an MROM instruction is encountered MROM unit 34 parses and issues the instruction into a subset of defined fast path instructions to effectuate the desired operation. MROM unit 34 dispatches *the subset of fast path instructions* to decode units 20.

Col. 5, lines 44-48 of Mahurin further teaches that "[w]hen MROM unit 34 detects such an instruction and the portion of the destination which is not updated by the instruction is detected as being required by the instruction, *a read of the destination will be done prior to execution of the instruction.*" The Examiner refers to this "read of the destination" as a "read operation." The Examiner *assumes* that the MROM unit 34 adds the read operation into the instruction stream prior to an executed instruction. Applicants respectfully submit that the Examiner's assumption is without merit.

As quoted above, the recited portion of Mahurin states that the "MROM unit 34 dispatches *the subset of fast path instructions* to decode units 20." Mahurin does not teach that the MROM unit 34 dispatches anything other than the subset of fast path instructions. Although the recited portion of Mahurin does state that "a read of the destination will be done prior to the execution of the instruction," Mahurin does not address how or where the

read operation is performed or what performs the read operation. Therefore, the Examiner's has absolutely no basis to assume that the MROM unit 34 adds the read operation into the instruction stream prior to an executed.

It should further be noted that Mahurin teaches that the "read operation" is done only *conditionally*. Col. 5, lines 50-54 of Mahurin teaches that "*when* MROM unit detects an instruction which updates only a portion of a destination and the portion of the destination which is not updated by the instruction is detected as not being required by the instruction, *[then] a read of the destination is not done*." Therefore, even assuming, *arguendo*, that Mahurin teaches "adding an extract instruction into an instruction stream before the first instruction," the present claims do not restrict this step to the conditional taught by Mahurin.

Throughout all of his rejections, the Examiner makes the flawed assumption that in Mahurin, the "read operation" is an "instruction" executed by the functional units 24 of Figure 1. That is, when Mahurin refers to "instructions," the Examiner assumes that this necessarily includes the "read operation." At no point does Mahurin refer to the "read operation" as an "instruction." At no point does Mahurin even state the function unts 24 perform the "read operation." Indeed, that Mahurin does *not* refer to the "read operation" as an "instruction" strongly indicates that Mahurin intends for the "read operation" to *not* be an "instruction." Therefore, the Examiner cannot simply interpret the "read operation" as an "instruction" in hindsight. Such an interpretation is clearly improper.

The citations to Isaman, Johnson and Hennessy do not cure the above problems.

Accordingly, the combination of Mahurin, Isaman, Johnson and Hennessy fails to teach or suggest "adding an extract instruction into an instruction stream before the first instruction," as claimed in claims 1, 8, 9 and 10.

Because the combination of Mahurin, Isaman, Johnson and Hennessy neither teaches nor suggests each and every element of claims 1, 8, 9 and 10, it is respectfully asserted that no *prima facie* case of obviousness has been made out.

(ii). The combination of Mahurin, Isaman, Johnson and Hennessy fails to teach or suggest "setting an indication that the result of said extract instruction needs to be written into a result field of the first instruction following the extract instruction," as claimed in claims 1, 8, 9 and 10.

The Examiner contends that Figure 3 and col. 14, line 52 to column 15, line 21 of Mahurin teaches "setting an indication that the result of said extract instruction needs to be written into a result field of the first instruction following the extract instruction. More particularly, the Examiner contends that "Dest Size signal 210 indicates to Select Logic to write the result of the read (extract) operation into the result register of the detected smaller instruction." Applicants respectfully disagree

Col. 15 lines 14-21 of Mahurin states the following regarding select logic 202:

> Select logic 202 provides select signals 240, 242, and 244 to multiplexors 204, 206 and 208 respectively. *Based on the destination size and location received upon bus 210*, the select signals will determine which portions of the data fed to each multiplexor will be gated out. It is this gating out of data from multiplexors 204, 206 and 208 which will form the final result to be conveyed from functional unit 24A upon bus 38a.

That is, the bus 210 provides only destination size and location information to the the select logic 202. Mahurin does not teach that any information regarding result of the "read operation" (which the Examiner contends is the claimed "extract instruction") is sent to the select logic 202. Further, Mahurin does not indicate that the executed instruction (which

the Examiner contends is the claimed "first instruction") even contains a "result field" as claimed in the instant claims. Therefore, the Examiner has no basis to assume that "Dest Size signal 210 indicates to Select Logic to *write the result of the read (extract) operation* into the result register of the detected smaller instruction."

The citations to Isaman, Johnson and Hennessy do not cure the above problems.

Accordingly, the combination of Mahurin, Isaman, Johnson and Hennessy fails to teach or suggest "setting an indication that the result of said extract instruction needs to be written into a result field of the first instruction following the extract instruction," as claimed in claims 1, 8, 9 and 10.

Because the combination of Mahurin, Isaman, Johnson and Hennessy neither teaches nor suggests each and every element of claims 1, 8, 9 and 10, it is respectfully asserted that no *prima facie* case of obviousness has been made out.

(iii). The combination of Mahurin, Isaman, Johnson and Hennessy fails to teach or suggest "writing the extract instruction result into the result field of said first instruction, and into all fields of operands being dependent of said first instruction," as claimed in claims 1, 8, 9 and 10.

The Examiner contends that col. 15, lines 14-40 of Mahurin teach "writing the extract instruction result into the result field of said instruction." Applicants respectfully disagree.

The recited portion of Mahurin teaches the *merging* of the results of instruction execution with "previous data," which is unrelated to the claimed invention. The recited portion of Mahurin does not refer to the read operation (which the Examiner contends is the claimed "extract instruction") at all. Further, the recited portion of Mahurin does not teach that the executed instruction (which the Examiner contends is the claimed "first

instruction") contains a "result field" as claimed in the instant application. Therefore, it is unclear exactly what the Examiner is contending is being written into.

Also, as previously noted, the "read operation"of Mahurin is performed *only if* the MROM unit 34 meets a conditional statement. This conditional is not present in the instant claims. Mahurin also teaches that when the read is performed, the contents resulting from the read "will subsequently be *merged* with the instruction result." (Mahurin, col. 5, lines 47-48) This indicates that after the "read operation" is performed, the contends resulting from the read operation are not kept. Therefore, if Mahurin does not even keep the contents resulting from the read, it does not following the such contents are written into "the result field of said first instruction," as claimed in claims 1, 8, 9 and 10.

The citations to Isaman, Johnson and Hennessy do not cure the above problems.

Accordingly, the combination of Mahurin, Isaman, Johnson and Hennessy fails to teach or suggest "writing the extract instruction result into the result field of said first instruction, and into all fields of operands being dependent of said first instruction," as claimed in claims 1, 8, 9 and 10.

Because the combination of Mahurin, Isaman, Johnson and Hennessy neither teaches nor suggests each and every element of claims 1, 8, 9 and 10, it is respectfully asserted that no *prima facie* case of obviousness has been made out.

(iv). The only suggestion for selectively piecing together the features of disparate elements in Mahurin, Isaman, Johnson and Hennessy, in the manner proposed by the Examiner stems from hindsight knowledge impermissibly derived from the Appellants' disclosure.

To support combining Mahurin, Isaman, Johnson and Hennessy, the Examiner makes numerous unsupported assumptions with regards to the references and relies on disparate elements from various references. It should be noted that the Examiner's motivations to combine are artificially created without any objective teaching in the prior art. As such, the only suggestion for selectively piecing together the features of these disparate elements in the manner proposed by the Examiner stems from hindsight knowledge impermissibly derived from the Apellants' disclosure.

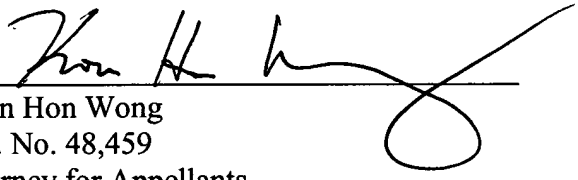
It should be noted that Isaman teaches exactly what is described in the instant application in Figure 6. Isaman simply replaces the "merge" instruction with the "stitch" instruction. However, the use of such a "stitch" instruction is expressly *rejected* by the instant application because of performance loss introduced by dependencies between the smaller bit length instruction, the merge (or stitch) instruction and the instruction consuming the target register.

The citations to Mahurin, Johnson and Hennessy do not cure the performance loss of Isaman.

D. CONCLUSION

Each and every element of the claimed invention is not described by the teachings of the applied prior art reference. The Examiner has failed to establish a *prima facie* case of obviousness of the presently claimed invention under 35 U.S.C. § 103(a) over the combination of Mahurin, Isaman, Johnson and Hennessy for at least the reasons noted above. Accordingly, it is respectfully requested that the Board reverse the rejection of claims 1-6, 8 and 9 under 35 U.S.C. § 102(e) and 35 U.S.C. § 103(a).

Respectfully submitted,

By: 

Koon Hon Wong
Reg. No. 48,459
Attorney for Appellants

F. Chau & Associates, LLC
130 Woodbury Road
Woodbury, NY 11797
TEL: (516) 692-8888
FAX: (516) 692-8889
Attorneys for Appellants

Claims Appendix

1. A method for operating a processor having an architecture of a larger bit-length with a program comprising instructions compiled to produce instruction results of at least one smaller bit-length, characterized by the steps of:

detecting when in program order a first instruction is to be dispatched which does not have a target register address as one of its sources, wherein the first instruction is one of the instructions compiled to produce instruction results of at least one smaller bit-length;

adding an extract instruction into an instruction stream before the first instruction, the extract instruction comprising the following steps of:

a. dispatching the extract instruction together with the following first instruction from an instruction queue into a Reservation Station;

b. issuing the extract instruction to an Instructional Execution Unit (IEU) as soon as all source operand data is available and an IEU is available according to respective issue scheme;

c. executing the extract instruction by an available IEU;

d. setting an indication that the result of said extract instruction needs to be written into a result field of the first instruction following the extract instruction, and;

e. writing the extract instruction result into the result field of said first instruction, and into all fields of operands being dependent of said first instruction.

2. The method according to claim 1 in which the step of writing the extract instruction result into the result field of said first instruction, is controlled by incrementing a tag specifying in which location the result of the first instruction has to be written.

3. The method according to claim 1, further comprising the steps of having the larger bit-length equal to 64-bit and the smaller bit-length equal to 32-bit.

4. The method according to claim 1, further comprising the steps of having the larger bit-length equal to 128-bit and the smaller bit-length equal to 64 or 32-bit.

5. The method according to claim 1, further comprising the step of when a second instruction is dependent of the first instruction, selectively inserting an extract instruction.

6. The method according to claim 1, further comprising the steps of dispatching said first instruction in the same cycle as the extract instruction, and assuring that in the same cycle both or none of said two instructions is written into a reservation station means, and in case of a multiple write into the same result latch reducing the respective multiple input signals for the latch by either one of an OR gate or an AND gate, respectively.

7. The method according to claim 1, further comprising the step of associating the same instruction execution unit for said first and said extract instruction.

8. A computer system having an out-of-order processing system for operating a processor having an architecture of a larger bit-length with a program comprising instructions compiled to produce instruction results of at least one smaller bit-length, said

computer system uses a computer readable machine language, said computer readable machine language comprises:

a first computer readable code for detecting when in program order a first instruction is to be dispatched which does not have a target register address as one of its sources, wherein the first instruction is one of the instructions compiled to produce instruction results of at least one smaller bit-length;

a second computer readable code for adding an extract instruction into an instruction stream before the first instruction, said extract instruction comprising additional computer readable code, said additional computer readable code comprises:

a third computer readable code for dispatching the extract instruction together with the following first instruction from the instruction queue into a Reservation Station,

a fourth computer readable code for issuing the extract instruction to an Instruction Execution Unit (IEU) as soon as all source operand data is available and an IEU is available according to respective issue scheme,

a fifth computer readable code for executing the extract instruction by an available IEU,

a sixth computer readable code for setting an indication that the result of said extract instruction needs to be written into a result field of the first instruction following the extract instruction, and

a seventh computer readable code for writing the extract instruction result into the result field of said first instruction, and into all fields of operands being dependent of said first instruction.

9. A computer system having an out-of-order processing system for operating a processor having an architecture of a larger bit-length with a program comprising instructions compiled to produce instruction results of at least one smaller bit-length, said computer system executes a readable machine language, said readable machine language comprises:

a first computer readable code for detecting when in program order a first instruction is to be dispatched which does not have a target register address as one of its sources, wherein the first instruction is one of the instructions compiled to produce instruction results of at least one smaller bit-length;

a second computer readable code for adding an extract instruction into an instruction stream before the first instruction, said extract instruction comprising additional computer readable code, said additional computer readable code comprises:

a third computer readable code for dispatching the extract instruction together with the following first instruction from the instruction queue into a Reservation Station,

a fourth computer readable code for issuing the extract instruction to an Instruction Execution Unit (IEU) as soon as all source operand data is available and an IEU is available according to respective issue scheme,

a fifth computer readable code for executing the extract instruction by an available IEU,

a sixth computer readable code for setting an indication that the result of said extract instruction needs to be written into a result field of the first instruction following the extract instruction, and

a seventh computer readable code for writing the extract instruction result into the result field of said first instruction, and into all fields of operands being dependent of said first instruction.

10. A processing system having an out-of-order processing system for operating a processor having an architecture of a larger bit-length with a program comprising instructions compiled to produce instruction results of at least one smaller bit-length, said processing system having means for executing a readable machine language, said readable machine language comprises:

a first computer readable code for detecting when in program order a first instruction is to be dispatched which does not have a target register address as one of its sources, wherein the first instruction is one of the instructions compiled to produce instruction results of at least one smaller bit-length;

a second computer readable code for adding an extract instruction into an instruction stream before the first instruction, said extract instruction comprising additional computer readable code, said additional computer readable code comprises:

a third computer readable code for dispatching the extract instruction together with the following first instruction from the instruction queue into a Reservation Station,

a fourth computer readable code for issuing the extract instruction to an Instruction Execution Unit (IEU) as soon as all source operand data is available and an IEU is available according to respective issue scheme,

a fifth computer readable code for executing the extract instruction by an available IEU,

a sixth computer readable code for setting an indication that the result of said extract instruction needs to be written into a result field of the first instruction following the extract instruction, and

a seventh computer readable code for writing the extract instruction result into the result field of said first instruction, and into all fields of operands being dependent of said first instruction.

Evidence Appendix

None

Related Procedings Appendix

None